

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-153029

(43)Date of publication of application : 10.06.1997

(51)Int.Cl. G06F 17/14
G06F 15/16

(21)Application number : 07-311224

(71)Applicant : FUJITSU LTD
UNIV AUSTRALIAN NATL

(22)Date of filing : 29.11.1995

(72)Inventor : NAKANISHI MAKOTO
MARKUS HOEGLUND

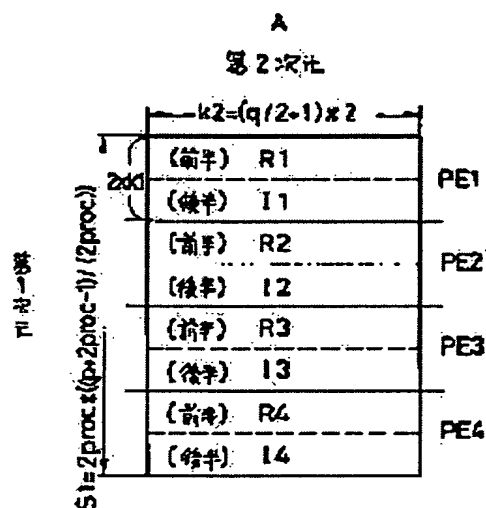
(54) MEMORY DISTRIBUTED PARALLEL COMPUTER FOR FAST FOURIER TRANSFORMATION AND ITS METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To efficiently execute fast Fourier transformation with a small data transfer volume.

SOLUTION: Two-dimensional real number data are stored in a two-dimensional array A with size $s1 \times k2$ extending over four processors PE1, PE2, PE3, and PE4. The size in the first dimension (the dimension of rows) of a partial array in each processor of the array A is an even number, and each partial array is divided into two sets of row vectors of the first half and the latter half. One row vector of the first half is regarded as the real part and one row vector in the latter half is regarded as the imaginary part to execute complex Fourier transformation in each processor, and the result of real Fourier transformation for the second dimension of each row vector is taken out.

Thereafter, the array A is transposed by parallel data transfer, and complex Fourier transformation for the first dimension is executed, and the result is transposed again to obtain the result of two-dimensional Fourier transformation. Fourier transformation for the first dimension and that for the second dimension of the array A can be executed in each processor in the closed form to efficiently perform the processing.



LEGAL STATUS

[Date of request for examination] 04.02.2000

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision
of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-153029

(43) 公開日 平成9年(1997)6月10日

(51) Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 17/14			G 0 6 F 15/332	A
15/16	3 9 0		15/16	3 9 0 Z

審査請求 未請求 請求項の数13 O L (全 22 頁)

(21) 出願番号	特願平7-311224	(71) 出願人	000005223 富士通株式会社 神奈川県川崎市中原区上小田中4丁目1番1号
(22) 出願日	平成7年(1995)11月29日	(71) 出願人	593094659 ジ オーストラリアン ナショナル ユニ バシティー オーストラリア国, エーシーティー, 0200, キャンベラ (番地なし)
		(72) 発明者	中西 誠 神奈川県川崎市中原区上小田中1015番地 富士通株式会社内
		(74) 代理人	弁理士 大菅 義之 (外1名) 最終頁に続く

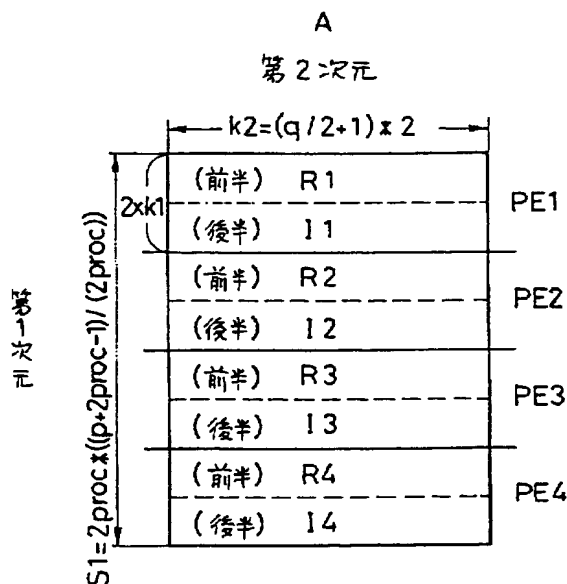
(54) 【発明の名称】 高速フーリエ変換を行うメモリ分散型並列計算機およびその方法

(57) 【要約】

【目的】 メモリ分散型並列計算機システムにおいて、少ないデータ転送で効率良く高速フーリエ変換を実行することを目的とする。

【構成】 2次元実数データを、4つのプロセッサ P E 1、P E 2、P E 3、P E 4にまたがる $s_1 \times k_2$ の大きさの2次元配列 A に格納する。配列 A の各プロセッサ内の部分配列の第1次元 (行の次元) の大きさは偶数であり、各部分配列は前半と後半の2組の行ベクトルに分割される。前半の1つの行ベクトルを実部、後半の1つの行ベクトルを虚部とみなして、各プロセッサ内で複素フーリエ変換を行い、各行ベクトルの第2次元についての実フーリエ変換の結果を取り出す。その後、配列 A を並列データ転送により転置して、第1次元についての複素フーリエ変換を行い、その結果を再び転置して2次元フーリエ変換の結果を求める。配列 A の第1次元および第2次元の各フーリエ変換を各プロセッサ内で閉じて行うことができ、処理が効率化される。

2次元配列の分割を示す図



【特許請求の範囲】

【請求項1】 データを複数のプロセッサのメモリに分散配置し、プロセッサ間でデータ転送を行いながら高速フーリエ変換を実行することのできるメモリ分散型並列計算機システムにおいて、

実数データを、第1次元と第2次元からなる第1の2次元配列として入力する入力手段と

前記第1の2次元配列の第1次元をプロセッサ数で分割して、第1次元が偶数であるような複数の部分配列を生成し、該複数の部分配列を前記複数のプロセッサに分散して記憶する配列記憶手段と、

前記複数の部分配列の各々を前記第1次元で2つに分割して得られる第1のデータと第2のデータのうち、該第1のデータを記憶する第1の記憶手段と、

前記第2のデータを記憶する第2の記憶手段と、

前記配列記憶手段から前記第1のデータおよび第2のデータを前記第1の記憶手段および第2の記憶手段に移し、該第1のデータを実部、該第2のデータを虚部とみなして前記第2次元についての複素フーリエ変換を各プロセッサ内で行い、該複素フーリエ変換の第1の変換結果を利用して、前記実数データのフーリエ変換の第2の変換結果を求める計算手段と、

前記第2の変換結果を出力する出力手段とを備えることを特徴とする並列計算機。

【請求項2】 2次元配列を転置する転置手段をさらに備え、前記計算手段は、前記第1の2次元配列の第2次元についての実フーリエ変換の第3の変換結果の一部を、前記第1の変換結果から求めて、該第3の変換結果の一部の実部および虚部からなる第2の2次元配列を前記配列記憶手段に格納し、前記転置手段は、該配列記憶手段に格納された該第2の2次元配列を転置して前記第1の記憶手段および第2の記憶手段に格納することにより、前記第3の変換結果の一部の実部および虚部をそれぞれ該第1の記憶手段および第2の記憶手段に格納し、前記計算手段は、該第1の記憶手段および第2の記憶手段のデータを用いて、該第2の2次元配列の第1次元についての複素フーリエ変換を各プロセッサ内で行うことを特徴とする請求項1記載の並列計算機。

【請求項3】 前記転置手段は、前記第2の2次元配列をブロック単位に分割して、プロセッサ間で該ブロック単位の並列データ転送を行うことにより、該第2の2次元配列を転置することを特徴とする請求項2記載の並列計算機。

【請求項4】 前記転置手段は、前記第2の2次元配列の第1次元についての複素フーリエ変換の結果得られる第3の2次元配列を再び転置し、前記第2の変換結果として前記配列記憶手段に格納することを特徴とする請求項2記載の並列計算機。

【請求項5】 前記入力手段は、前記実数データが1次元データするとき、該1次元データを前記第1の2次元配

列の形式に変換して入力することを特徴とする請求項1記載の並列計算機。

【請求項6】 2次元配列を転置する転置手段をさらに備え、前記計算手段は、前記第1の2次元配列の第2次元についての実フーリエ変換の第3の変換結果の一部を、前記第1の変換結果から求めて、該第3の変換結果の一部に回転因子を乗算し、乗算結果の実部および虚部からなる第2の2次元配列を前記配列記憶手段に格納し、前記転置手段は、該配列記憶手段に格納された該第2の2次元配列を転置して前記第1の記憶手段および第2の記憶手段に格納することにより、前記第3の変換結果の一部の実部および虚部をそれぞれ該第1の記憶手段および第2の記憶手段に格納し、前記計算手段は、該第1の記憶手段および第2の記憶手段のデータを用いて、該第2の2次元配列の第1次元についての複素フーリエ変換を各プロセッサ内で行い、前記第2の変換結果を求めることを特徴とする請求項5記載の並列計算機。

【請求項7】 データを複数のプロセッサのメモリに分散配置し、プロセッサ間でデータ転送を行いながら高速フーリエ変換を実行することのできるメモリ分散型並列計算機システムにおいて、

実数データを、第1次元と第2次元を有する第1の多次元配列として入力する入力手段と前記第1の多次元配列の第1次元をプロセッサ数で分割して、第1次元が偶数であるような複数の部分配列を生成し、該複数の部分配列を前記複数のプロセッサに分散して記憶する配列記憶手段と、

前記複数の部分配列の各々を前記第1次元で2つに分割して得られる第1のデータと第2のデータのうち、該第1のデータを記憶する第1の記憶手段と、

前記第2のデータを記憶する第2の記憶手段と、

前記配列記憶手段から前記第1のデータおよび第2のデータを前記第1の記憶手段および第2の記憶手段に移し、該第1のデータを実部、該第2のデータを虚部とみなして前記第2次元についての複素フーリエ変換を各プロセッサ内で行い、該複素フーリエ変換の第1の変換結果を利用して、前記実数データのフーリエ変換の第2の変換結果を求める計算手段と、

前記第2の変換結果を出力する出力手段とを備えることを特徴とする並列計算機。

【請求項8】 データを複数のプロセッサのメモリに分散配置し、プロセッサ間でデータ転送を行いながら高速フーリエ変換を実行することのできるメモリ分散型並列計算機システムを構成するプロセッサであって、第1次元と第2次元を有する多次元配列の形式で入力された実数データのうち、該第1次元をプロセッサ数で分割して割り当てられた部分データを記憶するための、第1次元が偶数であるような格納領域を有する配列記憶手段と、

前記格納領域に記憶された前記部分データを前記第1次

元で2つに分割して得られる第1のデータと第2のデータのうち、該第1のデータを記憶する第1の記憶手段と、

前記第2のデータを記憶する第2の記憶手段と、

前記配列記憶手段から前記第1のデータおよび第2のデータを前記第1の記憶手段および第2の記憶手段に移し、該第1のデータを実部、該第2のデータを虚部とみなして前記第2次元についての複素フーリエ変換を行い、該複素フーリエ変換の第1の変換結果を利用して、前記実数データのフーリエ変換を求める計算手段とを備えることを特徴とするプロセッサ。

【請求項9】 データを複数のプロセッサのメモリに分散配置し、プロセッサ間でデータ転送を行いながら高速フーリエ変換を実行することのできるメモリ分散型並列計算機システムにおいて、

長さ $N=p \times q$ の1次元データを $p \times q$ の第1の2次元配列とみなし、該第1の2次元配列を第1次元で分割して、各プロセッサ内の第1次元の大きさが偶数になるように分散配置する配列記憶手段と、

各プロセッサで第2次元のベクトルのペアを複素数とみなして、複素フーリエ変換を行い、変換結果から実フーリエ変換の $(q/2+1)$ 部分を計算した後、ローテーション演算を行う第1の計算手段と、

前記ローテーション演算の結果を、第1次元を分割配置した $(q/2+1) \times p$ の第2の2次元配列に転置する転置手段と、

該第2の2次元配列の長さ p の第2次元のベクトルに対する1次元複素フーリエ変換を各プロセッサで並列に行って、前記1次元データの実フーリエ変換の結果を求める第2の計算手段とを備えることを特徴とする並列計算機。

【請求項10】 データを複数のプロセッサのメモリに分散配置し、プロセッサ間でデータ転送を行いながら高速フーリエ変換を実行することのできるメモリ分散型並列計算機システムにおいて、

(p, q) の大きさの2次元データを $p \times q$ の第1の2次元配列に入力し、該第1の2次元配列を第1次元で分割して、各プロセッサ内の第1次元の大きさが偶数になるように分散配置する配列記憶手段と、

各プロセッサで第2次元のベクトルのペアを複素数とみなして、複素フーリエ変換を行い、変換結果から実フーリエ変換の $(q/2+1)$ 部分を計算する第1の計算手段と、

前記第1の計算手段による計算結果を、第1次元を分割配置した $(q/2+1) \times p$ の第2の2次元配列に転置する第1の転置手段と、

該第2の2次元配列の長さ p の第2次元のベクトルに対する1次元複素フーリエ変換を各プロセッサで並列に行う第2の計算手段と、

該第2の2次元配列の変換結果を $p \times (q/2+1)$ の

配列に転置して、前記2次元データの実フーリエ変換の結果のうち $(1:p, 1:q/2+1)$ の部分を求める第2の転置手段とを備えることを特徴とする並列計算機。

【請求項11】 データを複数のプロセッサのメモリに分散配置し、プロセッサ間でデータ転送を行いながら高速フーリエ変換を実行することのできるメモリ分散型並列計算機システムにおいて、

(p, q, r) の大きさの3次元データを $p \times q \times r$ の第1の3次元配列に入力し、該第1の3次元配列を第1次元で分割して、各プロセッサ内の第1次元の大きさが偶数になるように分散配置する配列記憶手段と、

各プロセッサで第3次元のベクトルのペアを複素数とみなして、複素フーリエ変換を行い、変換結果から実フーリエ変換の $(r/2+1)$ 部分を計算した後、第2次元のベクトルに対するフーリエ変換を各プロセッサで並列に行う第1の計算手段と、

前記第1の計算手段による計算結果を、第1次元を分割配置した $(r/2+1) \times q \times p$ の第2の3次元配列に転置する第1の転置手段と、

該第2の3次元配列の長さ p の第3次元のベクトルに対する1次元複素フーリエ変換を各プロセッサで並列に行う第2の計算手段と、

該第2の3次元配列の変換結果を $p \times q \times (r/2+1)$ の配列に転置して、前記3次元データの実フーリエ変換の結果のうち $(1:p, 1:q, 1:r/2+1)$ の部分を求める第2の転置手段と

を備えることを特徴とする並列計算機。

【請求項12】 データを複数のプロセッサのメモリに分散配置し、プロセッサ間でデータ転送を行いながら高速フーリエ変換を実行することのできるメモリ分散型並列計算機システムにおける記憶媒体であって、

実数データを、第1次元と第2次元を有する第1の多次元配列として入力する入力手段と前記第1の多次元配列の第1次元をプロセッサ数で分割して、第1次元が偶数であるような複数の部分配列を生成し、該複数の部分配列を前記複数のプロセッサに分散して配置する分散配置手段と、

前記複数の部分配列の各々を前記第1次元で2つに分割して第1のデータと第2のデータを生成し、該第1のデータを記憶する第1の記憶域と該第2のデータを記憶する第2の記憶域とを割り当てる割り当て手段と、

前記第1のデータおよび第2のデータを前記第1の記憶域および第2の記憶域に移し、該第1のデータを実部、該第2のデータを虚部とみなして前記第2次元についての複素フーリエ変換を各プロセッサ内で行い、該複素フーリエ変換の第1の変換結果を利用して、前記実数データのフーリエ変換の第2の変換結果を求める計算手段と、

前記第2の変換結果を出力する出力手段とを備えること

を特徴とする記憶媒体。

【請求項13】 メモリ分散型並列計算機システムにおいて、データを複数のプロセッサのメモリに分散配置し、プロセッサ間でデータ転送を行いながら高速フーリエ変換を実行する方法であって、

実数データを、第1次元と第2次元を有する第1の多次元配列として入力し、

前記第1の多次元配列の第1次元をプロセッサ数で分割して、第1次元が偶数であるような複数の部分配列を生成し、該複数の部分配列を前記複数のプロセッサに分散して配置し、

前記複数の部分配列の各々を前記第1次元で2つに分割して第1のデータと第2のデータを生成し、

該第1のデータおよび第2のデータを第1の記憶域および第2の記憶域に移し、

該第1のデータを実部、該第2のデータを虚部とみなして前記第2次元についての複素フーリエ変換を各プロセッサ内で行い、

該複素フーリエ変換の第1の変換結果を利用して、前記実数データのフーリエ変換の第2の変換結果を求めることを特徴とするフーリエ変換処理方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は高速フーリエ変換を行うメモリ分散型並列計算機システム、およびフーリエ変換処理方法に関する。

【0002】

【従来の技術とその問題点】 今日、科学技術計算等において、大規模な高速フーリエ変換（FFT：Fast Fourier Transformation）を計算するために、並列計算機システムがよく用いられている。FFTは、離散型フーリエ変換における項数がいくつかの因数に分解できる時、その性質を利用して必要な演算回数を削減した計算アルゴリズムである。特に、大規模な実フーリエ変換をVPP500等のベクトル分散並列計算機で行うとき、その性能を最大限に引き出すために、データ転送が少なく一括して計算できる方法が望まれている。

【0003】 従来の並列計算機によるFFT処理においては、フーリエ変換の度に、複数のプロセッシング・エレメント（以後、PE、またはプロセッサと記す）に分散したデータをアクセスする必要がある。このため、多量のデータ転送が行われ、処理効率が低下するという問題が生じる。

【0004】 実数のフーリエ変換を行うには、複素フーリエ変換の虚部を0として計算する方法と、2つの実数をそれぞれ複素数の実部、虚部とみなして、複素フーリエ変換を行う方法の2つがある。後者は、前者に比べて使用するメモリ量が約半分で済むという利点がある。これは、フーリエ変換した結果の複素数の中に共役関係を持つペアがあり、求めるべきデータを約半分に減らすこ

とができることを利用したためである。

【0005】 実際には、変換される実数を複数の列に並べて、偶数列と奇数列（または偶数行と奇数行）のデータをそれぞれ実部および虚部とみなして複素フーリエ変換を行っており、最後に、その結果から必要な約半分の実フーリエ変換の結果を作り出している。残りの半分の結果は、共役関係を利用して求めることができる。しかしながら、偶数列と奇数列（または偶数行と奇数行）のデータは必ずしも同じプロセッサに格納されるとは限らず、フーリエ変換のアクセスパターンが全プロセッサにまたがるため、多量のデータ転送が必要となる。

【0006】 並列計算機においてはデータ転送に伴うオーバーヘッドが大きいと、このような多量のデータ転送が生じると、ベクトル処理および並列処理の効率が悪くなる。

【0007】 本発明は、少ないデータ転送で効率良く実FFTを実行することのできるメモリ分散型並列計算機システムと、その方法を提供することを目的とする。

【0008】

【問題を解決するための手段】 図1は、本発明の並列計算機システムの原理図である。図1の並列計算機システムは、データを複数のプロセッサのメモリに分散配置し、プロセッサ間でデータ転送を行いながら高速フーリエ変換を実行することのできるメモリ分散型並列計算機システムであって、入力手段1、配列記憶手段2、第1の記憶手段3、第2の記憶手段4、計算手段5、転置手段6、および出力手段7を備える。

【0009】 入力手段1は、実数データを、第1次元と第2次元からなる第1の2次元配列として入力する。配列記憶手段2は、第1の2次元配列の第1次元をプロセッサ数で分割して、第1次元が偶数であるような複数の部分配列を生成し、それらの複数の部分配列を上記複数のプロセッサに分散して記憶する。

【0010】 第1の記憶手段3は、上記複数の部分配列の各々を第1次元で2つに分割して得られる第1のデータと第2のデータのうち、第1のデータを記憶し、第2の記憶手段4は第2のデータを記憶する。

【0011】 計算手段5は、配列記憶手段2から第1のデータおよび第2のデータを第1の記憶手段3および第2の記憶手段4に移し、第1のデータを実部、第2のデータを虚部とみなして第2次元についての複素フーリエ変換を各プロセッサ内で行い、その複素フーリエ変換の第1の変換結果を求める。そして、第1の変換結果を利用して、上記実数データのフーリエ変換の第2の変換結果を求める。

【0012】 出力手段7は、上記第2の変換結果を出力する。また、計算手段5は、上記第1の2次元配列の第2次元についての実フーリエ変換の第3の変換結果の一部を、上記第1の変換結果から求めて、その第3の変換結果の一部の実部および虚部からなる第2の2次元配列

を配列記憶手段2に格納する。このとき、転置手段6は、配列記憶手段2に格納された第2の2次元配列を転置して第1の記憶手段3および第2の記憶手段4に格納することにより、上記第3の変換結果の一部の実部および虚部をそれぞれ第1の記憶手段3および第2の記憶手段4に格納する。そして、計算手段5は、第1の記憶手段3および第2の記憶手段4のデータを用いて、第2の2次元配列の第1次元についての複素フーリエ変換を各プロセッサ内で行い、その結果から上記第2の変換結果を求める。

【0013】例えば、図1の入力手段1と出力手段7は、実施例における図2の入出力装置13に対応し、配列記憶手段2、第1の記憶手段3、第2の記憶手段4は、各プロセッサ11-1、11-2、・・・、11-M内のメモリ16に対応する。また、計算手段5と転置手段6は、各プロセッサ11-1、11-2、・・・、11-M内の処理部15に対応する。さらに、例えば、これらの各手段の機能をプログラム化して記憶したディスク装置等の記憶媒体もまた、これらの各手段に対応する。

【0014】

【作用】入力手段1が入力した2次元実数データが、第1の2次元配列として、配列記憶手段2により複数のプロセッサに分散配置される。2次元実フーリエ変換を行うには、この第1の2次元配列を第1次元および第2次元についてそれぞれフーリエ変換すればよい。このとき、第1の2次元配列の第1次元の大きさを、例えば(2×プロセッサ数)の倍数とし、それをプロセッサ数で分割したときに、各プロセッサ内の部分配列の第1次元が偶数になるようにしておく。これにより、各部分配列を第1次元で2つに分割して格納することが可能になる。第1の記憶手段3は、各プロセッサ毎に、各部分配列を分割した2つデータのうち第1のデータを記憶し、第2の記憶手段4は第2のデータを記憶する。

【0015】計算手段5は、第1の記憶手段3内の第1のデータを複素数の実部とみなし、第2の記憶手段4内の第2のデータを虚部とみなして、各プロセッサ内で複素数のフーリエ変換を行うことができる。これにより、分割されていない第2次元についての複素フーリエ変換が各プロセッサ内で行われ、第1の変換結果が得られる。第1のデータおよび第2のデータのそれぞれの第2次元についての実フーリエ変換の結果は、第1の変換結果から容易に得ることができ、次に第1次元についての複素フーリエ変換を行えば、上記実数データの2次元フーリエ変換の結果である第2の変換結果が得られる。

【0016】このように、入力された2次元データを各プロセッサ内で2つの部分に分け、一方を実部、他方を虚部とみなすことにより、2組の実数データをまとめてフーリエ変換することができる。このとき、計算は各プロセッサ内で閉じて行われるため、データ転送に要する

時間が節約され、処理が効率化される。

【0017】尚、入力データが1次元実数データの場合は、これを2次元データに変換して配列記憶手段2に記憶させることにより、2次元データと同様に処理することが可能である。この場合は、第1次元についての複素フーリエ変換を行う前に、各プロセッサ内でローテーションの計算を行えばよい。また、3次元以上の次元の実数データのフーリエ変換についても、基本的には2次元データと同様の手法により処理することができる。

【0018】また、第1の変換結果から第2の変換結果を得る時、計算手段5は、まず第1の2次元配列の第2次元についての実フーリエ変換の結果である第3の変換結果を、第1の変換結果から求める。このとき、第3の変換結果には一般に特定の共役関係があり、これを利用すれば、第3の変換結果の約半分は他の部分から計算できることが知られている。したがって、第3の変換結果については、その一部分(約半分)のみを求めればよい。求めた第3の変換結果の一部の実部および虚部は、第2の2次元配列として配列記憶手段2に格納される。

【0019】しかし、このままでは、第2の2次元配列の第1次元が複数のプロセッサにまたがって配置されているため、残された第1次元についてのフーリエ変換を各プロセッサ内で閉じて行うことができない。そこで、転置手段6が第2の2次元配列を転置して、第1の記憶手段3および第2の記憶手段4に格納する。これにより、第3の変換結果の一部の実部および虚部が、それぞれ第1の記憶手段3および第2の記憶手段4に格納される。計算手段は、これらを1組の複素数として、第2の2次元配列の第1次元についての複素フーリエ変換を各プロセッサ内で行い、最終的に第2の変換結果を求める。得られた第2の変換結果は、入力された2次元実数データのフーリエ変換の結果を与えている。

【0020】このように、第2次元についてのフーリエ変換の後に2次元配列を転置すれば、第1次元についてのフーリエ変換も複数のプロセッサで並列に処理することができる。1次元実数データの場合は、配列記憶手段2に格納された第2の2次元配列に対してローテーションの計算が行われ、その結果が転置されて、第1次元についてのフーリエ変換が行われる。

【0021】

【実施例】以下、図面を参照しながら本発明の実施例を詳細に説明する。図2は、実施例における並列計算機システムの構成図である。図2の並列計算機システムは、ネットワーク12により結合された複数のプロセッサ11-1、11-2、・・・、11-Mからなり、ネットワーク12に接続された入出力装置13を備える。ネットワーク12は、任意の2つのプロセッサ間でデータ転送を行えるように設計されており、例えばクロスバー・ネットワークである。入出力装置13は、例えばディスプレイやキーボードを備えた端末装置であり、フーリエ

変換されるデータを入力し、変換結果を出力する。

【0022】また、各プロセッサ11-1、11-2、
・・・、11-Mは、それぞれ通信部14、処理部1
5、メモリ16、およびそれらを接続する内部バス17
を備える。通信部14は、ネットワーク12を介して、
他のプロセッサや入出力装置13との間でデータ転送を
行う。メモリ16は、各プロセッサに割り当てられたデ
ータや他のプロセッサから転送されたデータを記憶す
る。処理部15は、メモリ16に記憶されたデータを用

$$f_j = \sum_{n=0}^{N-1} x_n \omega_N^{nj} \quad \dots (1)$$

【0026】ただし、

$$\omega_N \equiv \exp(-2\pi i/N) \quad \dots (2)$$

とする。今、 $N=p \cdot q$ と因数分解できるとすると、

$$n = n_1 + n_2 \cdot p, \quad \dots (3)$$

$$(n_1 = 0, 1, \dots, p-1, n_2 = 0, 1, \dots, q-1)$$

$$j = j_1 + j_2 \cdot q, \quad \dots (4)$$

$$(j_1 = 0, 1, \dots, q-1, j_2 = 0, 1, \dots, p-1)$$

$$x(n_1, n_2) = x_n, \quad \dots (5)$$

$$f(j_1, j_2) = f_j \quad \dots (6)$$

とおくことができる。(3)、(4)、(5)、(6) 【0027】

式を(1)式に代入して、 n についての和を n_1 と n_2 【数2】

についての和に書き換えると、

$$\begin{aligned} f(j_1, j_2) &= \sum_{n_1=0}^{p-1} \omega_p^{n_1 j_2} \omega_N^{n_1 j_1} \\ &\cdot \sum_{n_2=0}^{q-1} x(n_1, n_2) \omega_q^{n_2 j_1} \end{aligned} \quad \dots (7)$$

【0028】となる。このように、離散フーリエ変換の
項数 N が適当な因数に分解できるときには、1次元のデ
ータを2次元データとみなして処理することができる。
並列計算機による処理においては、例えば $x(n_1, n_2)$ の第1次元を複数のプロセッサに分割して配置す

いて、演算等の処理を行う。

【0023】本実施例においては、図2の並列計算機シ
ステムにより行われる1次元、2次元、および3次元の
フーリエ変換処理について順に説明する。最初に、フー
リエ変換に必要なとなる数学的な前提について述べる。

【0024】1次元の離散フーリエ変換は次式で与えら
れる。

【0025】

【数1】

る。このとき、(7)式は次の4段の処理に分けられ
る。

【0029】

【数3】

$$(1 \text{ 段}) \quad Y_1(n_1, j_1) = \sum_{n_2=0}^{Q-1} x(n_1, n_2) \omega_q^{n_2 j_1} \quad \dots (8)$$

$$(2 \text{ 段}) \quad Y_2(n_1, j_1) = \omega_N^{n_1 j_1} Y_1(n_1, j_1) \quad \dots (9)$$

$$(3 \text{ 段}) \quad Y_3(j_1, n_1) = Y_2(n_1, j_1) \quad \dots (10)$$

$$(4 \text{ 段}) \quad Y_4(j_1, j_2) = \sum_{n_1=0}^{P-1} Y_3(j_1, n_1) \omega_p^{n_1 j_2} \quad \dots (11)$$

【0030】(8)式は2次元配列 $x(n_1, n_2)$ の添字 n_2 についてのフーリエ変換を表し、(9)式は $Y_1(n_1, j_1)$ に回転因子 $\omega_N^{n_1 j_1}$ を乗算する計算(ローテーション)を表し、(10)式は2次元配列 $Y_2(n_1, j_1)$ の転置を表し、(11)式は $Y_3(j_1, n_1)$ の添字 n_1 についてのフーリエ変換を表す。第3段で転置を行うのは、常に2次元配列の第2次元についてのフーリエ変換を行うようにするためである。こ

$$z_n = x_1 n + i x_2 n$$

とおく。また、 z_n 、 $x_1 n$ 、 $x_2 n$ のフーリエ変換の結果を、それぞれ $\{\alpha^z_j\}$ 、 $\{\alpha^{x1}_j\}$ 、 $\{\alpha^{x2}_j\}$ とおくと、

ここでは、2次元配列の第2次元は分割されていないので、第2次元についてのフーリエ変換は各プロセッサ内で独立に行うことができる。

【0031】次に、実1次元フーリエ変換を複素フーリエ変換と組み合わせて行う方法について説明する。変換すべき実データを $\{x_1 n\}$ と $\{x_2 n\}$ ($n=0, \dots, Q-1$)とに分け、それぞれを複素数の実部と虚部とみなして、

$$\dots (12)$$

【0032】

【数4】

$$\alpha^z_j = \sum_{n=0}^{Q-1} z_n \omega_q^{n j} \quad \dots (13)$$

$$\alpha^{x1}_j = \sum_{n=0}^{Q-1} x_1 n \omega_q^{n j} \quad \dots (14)$$

$$\alpha^{x2}_j = \sum_{n=0}^{Q-1} x_2 n \omega_q^{n j} \quad \dots (15)$$

【0033】となる。(14)、(15)式の α^{x1}_j 、 α^{x2}_j は、(13)式の α^z_j を用いて、次のように表される。

【0034】

【数5】

$$\alpha^{x1}_j = (\alpha^z_j + (\alpha^{z_{Q-j}})^*) / 2 \quad \dots (16)$$

$$\alpha^{x2}_j = (\alpha^z_j - (\alpha^{z_{Q-j}})^*) / 2i \quad \dots (17)$$

$$\alpha^{x1}_0 = \text{Re}(\alpha^z_0) \quad \dots (18)$$

$$\alpha^{x2}_0 = \text{Im}(\alpha^z_0) \quad \dots (19)$$

【0035】ここで、 α^* は α の複素共役を表し、 Re

(α)、 $\text{Im}(\alpha)$ はそれぞれ α の実部、虚部を表す。

したがって、 $\{\alpha^z j\}$ が求めれば、(16)、(17) 式に従って $\{\alpha x1_j\}$ 、 $\{\alpha x2_j\}$ を計算すること

$$\alpha x1_{Q-j} = (\alpha x1_j)^*, \quad j=1, \dots, Q-1 \quad \dots (20)$$

$$\alpha x2_{Q-j} = (\alpha x2_j)^*, \quad j=1, \dots, Q-1 \quad \dots (21)$$

なる共役関係があるため、すべての j について $\alpha x1_j$ 、 $\alpha x2_j$ を求める必要はない。実際には、最初の $(Q/2 + 1)$ 個の $\alpha x1_j$ と $\alpha x2_j$ について、実フーリエ変換の結果を求めれば充分である。ただし、 $Q/2$ は Q を 2 で割った時の整数商（剰余は切り捨て）を表す。以下、特に断らない限り、数式に現れる除算は整数商を表すものとする。

【0036】以上の知識を前提として、1次元実FFTの方法を説明する。図3は、 $N=p \cdot q$ と分解できる N 個の実数を(1)式に従って変換する1次元実FFTのフローチャートである。図3において処理が開始される

$$s1 = 2 \times \text{proc} \times k1, \quad \dots (22)$$

$$k1 = (p + 2 \times \text{proc} - 1) / (2 \times \text{proc}) \quad \dots (23)$$

と定め、第1次元を proc 台のプロセッサにより均等に分割したとき、各プロセッサに格納される部分の行数

$$k2 = (q/2 + 1) \times 2$$

以上とする。図4では、 A の第2次元の大きさは $k2$ となっており、配列 A は $A(s1, k2)$ と表される。次

$$k3 = ((q/2 + \text{proc}) / \text{proc}) \times \text{proc} \quad \dots (25)$$

として、 A の転置用の格納配列 $B(2 \times k3, s1)$ を用意する。 B は同じ大きさの2つの配列 $BR(k3, s1)$ と $BI(k3, s1)$ からなる。 BR 、 BI の第1次元の大きさ $k3$ は、 A の第2次元の大きさ $k2$ の半分

$$2 \times k3 \geq k2 = (q/2 + 1) \times 2$$

なる関係がある。

【0038】そして、 $A(s1, k2)$ 、 $BR(k3, s1)$ 、 $BI(k3, s1)$ の第1次元を各プロセッサで均等に分割し、分割された各プロセッサの部分 $a(2 \times k1, k2)$ 、 $br(k3/\text{proc}, s1)$ 、 $bi(k3/\text{proc}, s1)$ とする。

【0039】次に、各プロセッサ内で、(8)式に対応する1段目の処理を並列に行う。ただし、このときメモリの使用領域を節約するために、各プロセッサに割り当てられた行ベクトルを前半と後半の2組に分け、一方を実部、他方を虚部として、複素数のフーリエ変換を行う。

【0040】まず、各プロセッサは行ベクトルのペアを作り、一方を実部、他方を虚部として別々の領域に格納する(ステップS2)。具体的には、格納領域 br 、 bi をそれぞれ格納配列 $br(k1, 2 \times k3)$ 、 $bi(k1, 2 \times k3)$ として利用し、 $a(1:k1, k2)$ を $br(k1, 1:k2)$ にコピーし、 $a(k1+1:2 \times k1, k2)$ を $bi(k1, 1:k2)$ にコピーする。

【0041】図4では、4台のプロセッサ $PE1$ 、 $PE2$ 、 $PE3$ 、 $PE4$ の各々に割り当てられた $a(2 \times k$

ができるが、実フーリエ変換の結果には、

と、まず入出力装置13は与えられた1次元データを取り込み、それを $N=p \times q$ の2次元配列とみなして、各プロセッサ(PE)に所定数の行ベクトルを均等に割り当てる(ステップS1)。割り当てられた行ベクトルは、各プロセッサのメモリ16に格納される。

【0037】図4は、このときの2次元データを格納する2次元格納配列 A の分割方法を示している。図4では、簡単のためプロセッサ数を4($PE1$ 、 $PE2$ 、 $PE3$ 、 $PE4$)としているが、より一般的にはそれを proc と書くことにする。このとき、 A の第1次元(行の次元)の大きさ $s1$ を、

が偶数になるようにする。また、 A の第2次元の大きさは、

$$\dots (24)$$

に、

を proc 台のプロセッサで分割できるように、 $k2/2 = (q/2 + 1)$ を修正したものである。したがって、

$$\dots (26)$$

$1, k2$ が、前半部分と後半部分に分割されている。 $PE1$ のデータは $R1$ と $I1$ に分けられ、 $PE2$ のデータは $R2$ と $I2$ に分けられ、 $PE3$ のデータは $R3$ と $I3$ に分けられ、 $PE4$ のデータは $R4$ と $I4$ に分けられる。そして、これらのデータが、図5に示すように、配列 BR 、 BI にコピーされる。図5において、 $R1$ 、 $R2$ 、 $R3$ 、 $R4$ は実部として BR に格納され、 $I1$ 、 $I2$ 、 $I3$ 、 $I4$ は虚部として BI に格納されている。このとき、 BR 、 BI は、それぞれ $BR(s1/2, 2 \times k3)$ 、 $BI(s1/2, 2 \times k3)$ なる配列として利用される。

【0042】次に、 $br(k1, 1:k2)$ の行ベクトルと、対応する $bi(k1, 1:k2)$ の行ベクトルとを束ねて、各プロセッサで複素数のフーリエ変換を行う(ステップS3)。つまり、 $br(k1, 1:k2)$ の1つの行ベクトルの各要素を(12)式の $x1_n$ とみなし、対応する $bi(k1, 1:k2)$ の行ベクトルの要素を(12)式の $x2_n$ とみなして、複素数 z_n に対するフーリエ変換を(13)式により計算する。ただし、このとき $Q=q$ とする。

【0043】そして、(16)～(19)式により、最初の $(q/2 + 1)$ 個の実フーリエ変換の結果を求め、

その実部および虚部をそれぞれ $a(2 \times k1, 1 : q/2 + 1)$ および $a(2 \times k1, q/2 + 2 : k2)$ に格納する(ステップS4)。図6は、このときの格納方法を示している。図6において、Aの第2次元の前半が実部に割り当てられ、後半が虚部に割り当てられていることが分かる。例えばプロセッサPE1内では、図4のR1部分の行ベクトルの実フーリエ変換結果のうち、最初の $(q/2 + 1)$ 個が実際に求められ、それらの実部がCR1に格納され、虚部がDR1に格納される。また、図4のI1部分の行ベクトルの実フーリエ変換結果のうち、最初の $(q/2 + 1)$ 個が実際に求められ、それらの実部がCI1に格納され、虚部がDI1に格納される。他のプロセッサの領域CR2、DR2、CI2、DI2、CR3、DR3、CI3、DI3、CR4、DR4、CI4、DI4についても同様である。

【0044】図4、図5のような格納方法によれば、変換される複素数の実部と虚部が同じプロセッサ内にあるので、複素フーリエ変換を各プロセッサ内で独立に行うことができ、余分なデータ転送を行う必要がない。また、長さqの実データのペアに対して、一方を実部、他方を虚部とみなして複素フーリエ変換を行った結果から、 $(q/2 + 1)$ 個の実フーリエ変換の結果を計算する方法をベースにして、1次元のフーリエ変換を束ねて行うことで、プロセッサのベクトル性能を引き出すことができる。

【0045】次に、各プロセッサ内で(9)式に対応する2段目の処理を並列に行って、ローテーションを計算する(ステップS5)。この計算結果は再び配列Aに格納される。

【0046】この後、(10)式に対応する3段目の処理を行って、配列Aの $p \times (q/2 + 1) \times 2$ のデータをブロック(小領域)に分割し、ブロックレベルで転置して配列Bに格納する(ステップS6)。このとき、格納領域BR、BIはそれぞれ配列BR($k3, s1$)、BI($k3, s1$)として利用され、その第1次元に関して分割されている。そして、プロセッサ間のデータ転送により、A($p, 1 : q/2 + 1$)のデータがBR($1 : q/2 + 1, p$)に転置され、A($p, q/2 + 2 : (q/2 + 1) \times 2$)のデータがBI($1 : q/2 + 1, p$)に転置される。

【0047】図7は、ステップS6で行われる行列の転置処理のフローチャートである。図7を参照しながら、配列Aの実部を配列BRに転置する処理について説明する。図7において処理が開始されると、まず配列Aの実部、配列BRをメッシュで区切る(ステップS11)。これにより、各プロセッサ内にあるAの実部とBRの部分が、それぞれプロセッサ数 $proc$ に相当する数のブロックに分割される。図8は、転置前の配列Aの実部(または虚部)の分割例を示しており、図9は、転置後のデータを格納する配列BR(またはBI)の分割例を

示している。

【0048】図8において、Aの実部のうちプロセッサPE1内にあるデータは、A11、A12、A13、A14の4つのブロックに分割されている。同様に、プロセッサPE2内のデータはブロックA21、A22、A23、A24に分割され、プロセッサPE3内のデータはブロックA31、A32、A33、A34に分割され、プロセッサPE4内のデータはブロックA41、A42、A43、A44に分割される。また、図9において、BRのうちプロセッサPE1内にあるデータは、B11、B12、B13、B14の4つのブロックに分割されている。同様に、プロセッサPE2内のデータはブロックB21、B22、B23、B24に分割され、プロセッサPE3内のデータはブロックB31、B32、B33、B34に分割され、プロセッサPE4内のデータはブロックB41、B42、B43、B44に分割される。

【0049】次に、各プロセッサで、 $K = (\text{そのプロセッサの番号})$ 、 $J = K$ 、 $\#ct = 1$ とおき(ステップS12)、ブロック A_{JK} の転置行列 A_{JK}^T を求めて、ブロック B_{KJ} に格納する(ステップS13)。ここで、 B_{KJ} が同じプロセッサ内にある場合は A_{JK}^T をそこに格納し、 B_{KJ} が他のプロセッサ内にあるときはデータ転送を行う。そして、 $K = \text{mod}(K, proc) + 1$ 、 $\#ct = \#ct + 1$ とおき(ステップS14)、 $\#ct$ と $proc$ を比較する(ステップS15)。ここで、 $\text{mod}(K, proc)$ は、 K を $proc$ で割った時の剰余を意味する。 $\#ct$ が $proc$ を超えていなければステップS13以降の処理を繰り返し、 $\#ct$ が $proc$ を超えると処理を終了する。

【0050】例えば、プロセッサPE1の場合は、まず最初に A_{11}^T を求めて同じプロセッサ内の B_{11} に格納する(ステップS13)。次に、 $K = \text{mod}(1, 4) + 1 = 2$ となるので(ステップS14)、 A_{12}^T を求める(ステップS13)。ところが、対応する格納先の B_{21} はプロセッサPE2内にあるため、 A_{12}^T をプロセッサPE2に転送する。同様に、 A_{13}^T 、 A_{14}^T を順次求めて、プロセッサPE3の B_{31} 、PE4の B_{41} にそれぞれ転送する。すべてのプロセッサが同様の処理を行った結果、BRの内容は図10に示ようになる。図10において、プロセッサPE1内には A_{11}^T 、 A_{21}^T 、 A_{31}^T 、 A_{41}^T が格納され、プロセッサPE2内には A_{12}^T 、 A_{22}^T 、 A_{32}^T 、 A_{42}^T が格納され、プロセッサPE3内には A_{13}^T 、 A_{23}^T 、 A_{33}^T 、 A_{43}^T が格納され、プロセッサPE4内には A_{14}^T 、 A_{24}^T 、 A_{34}^T 、 A_{44}^T が格納されている。

【0051】ステップS13においては、 $proc$ 台のプロセッサの間でデータ転送が発生するが、VPP500のように各プロセッサに対して同時に読み込みと書き込みができる並列計算機では、配列の対角方向に並ぶブロック要素について並列にデータ転送を行うことができる。例えば、図8の斜線部分のブロック A_{12} 、 A_{23} 、 A_{34} 、

A_{41} の転置データ A_{12}^T 、 A_{23}^T 、 A_{34}^T 、 A_{41}^T は、図9の斜線部分の各ブロックに並列に転送される。このような並列転置処理を行うことにより、データ転送のコストが軽減される。

【0052】配列Aの虚部を配列B1に転置する場合も、同様の処理を行う。このとき、Aの虚部は図8のように分割され、B1は図9のように分割される。ここで、配列Aを転置しておくことにより、配列Aの列ベクトルに対するフーリエ変換を配列Bの行ベクトルに対するフーリエ変換に置き換えることができ、次の4段目の処理を各プロセッサで並列に行うことができる。

【0053】そして、各プロセッサで(11)式に対応する4段目の処理を並列に行って(ステップS7)、処理を終了する。ステップS7では、各プロセッサは行列の転置結果を用いて、メモリ16内の長さpの各行ベクトルに対する複素FFTを行う。これにより、与えられた $N=p \times q$ 個の実数のフーリエ変換の結果のうち、 $p \times (q/2+1)$ 個が得られる。残りの $p \times (q - (q/2+1))$ 個の実フーリエ変換の値は得られた結果と共役関係にあるため、これですべての結果を求めたことになる。

【0054】次に、図11から図16までを参照しながら、1次元実FFTの具体例について説明する。図11は、入力データを2次元の格納配列に収容して4台のプロセッサに分散配置した例を示している。図11では、 $N=300$ 個の実数が1次元データとして入力され、 $p=20$ 、 $q=15$ として、 20×15 個の2次元データ $x(n1, n2)$ ($n1=1, 2, \dots, 20$, $n2=1, 2, \dots, 15$)として分散配置される(ステップS1)。このとき、(22)、(23)、(24)、(25)式より $s1=24$ 、 $k1=3$ 、 $k2=16$ 、 $k3=8$ となり、 20×15 個のデータを格納する配列Aの大きさは $s1 \times k2=24 \times 16$ となる。また、Aの第1次元を4台のプロセッサで分割するので、1つのプロセッサに割り当てられる格納配列の大きさは 6×16 となり、その行ベクトルの数(6)は偶数になる。ここでは、プロセッサPE1、PE2、PE3のすべての行ベクトルとプロセッサPE4の2本の行ベクトルにデータが格納されている。そして、配列Aのデータを格納しない部分の要素(図11の*印)の値は、必要のないオーバーフローやアンダーフローを避けるために0にしておく。

【0055】こうして配列Aに格納されたデータは、各プロセッサ内で別の格納配列Bにコピーされる(ステップS2)。図12は、図11のデータがコピーされた配列Bを示している。ここでは、配列Bは $s1 \times (2 \times k3)=24 \times 16$ の配列として利用され、それぞれ 12×16 の大きさを持つ配列BRとB1に分けられる。そして、それらのうちのそれぞれ 3×16 の部分が各プロセッサに配置される。各プロセッサに割り当てられたA

の6本の行ベクトルのうち、上半分の3本はBRの対応する領域にコピーされ、下半分の3本はB1の対応する領域にコピーされる。

【0056】次に、各プロセッサ内で、BRに格納されたデータを実部とみなし、B1内でそれと同じ位置に格納されたデータを虚部とみなして複素数をつくり、そのフーリエ変換を計算する(ステップS3)。例えば、プロセッサPE1内の1行目の行ベクトルについては、 $x10=x(1, 1)$ 、 $x11=x(1, 2)$ 、 \dots 、 $x114=x(1, 15)$ 、かつ、 $x20=x(4, 1)$ 、 $x21=x(4, 2)$ 、 \dots 、 $x214=x(4, 15)$ として、(12)式に代入し、15個の複素数 z_n を作る。そして、 $Q=15$ として(13)式によりそのフーリエ変換を求め、その結果得られる複素数 αz_j ($j=0, 1, 2, \dots, 14$)の実部をBRに格納し、虚部をB1に格納する。他の行ベクトルについても同様である。

【0057】図13は、フーリエ変換の結果を格納した配列BR、B1を示している。図13において、BR内の $X(n1, n2)$ ($n1=1, 2, 3, 7, 8, 9, 13, 14, 15, 19, 20$, $n2=1, \dots, 15$)は、各フーリエ変換結果の実部を表し、B1内の $X(n1, n2)$ ($n1=4, 5, 6, 10, 11, 12, 16, 17, 18$, $n2=1, \dots, 15$)は、各フーリエ変換結果の虚部を表している。例えば、図12の $x(1, n2)$ を実部、 $x(4, n2)$ を虚部として変換した結果得られた複素数の実部が $X(1, n2)$ 、虚部が $X(4, n2)$ である。

【0058】次に、BR、B1に格納された複素フーリエ変換の結果を利用して、(16)～(19)式により実フーリエ変換の結果 $\alpha x1_j$ 、 $\alpha x2_j$ を求め、配列Aに格納する(ステップS4)。ただし、この場合は(20)、(21)式の共役関係があるので、jの最初の $15/2+1=8$ 個の値について $\alpha x1_j$ 、 $\alpha x2_j$ を求めればよい。こうして求められた $x(n1, n2)$ の $n2$ に関するフーリエ変換の結果を $\alpha(n1, j2)$ ($n1=1, \dots, 20$, $j2=1, \dots, 8$)と書くことにする。例えば、 $x(1, 1)$ 、 $x(1, 2)$ 、 \dots 、 $x(1, 15)$ のフーリエ変換の結果のうち、約半分に当たる $\alpha(1, 1)$ 、 $\alpha(1, 2)$ 、 \dots 、 $\alpha(1, 8)$ が求められ、同様に、 $x(4, 1)$ 、 $x(4, 2)$ 、 \dots 、 $x(4, 15)$ のフーリエ変換の結果のうち、約半分に当たる $\alpha(4, 1)$ 、 $\alpha(4, 2)$ 、 \dots 、 $\alpha(4, 8)$ が求められる。これらの実フーリエ変換の結果は、図14に示すように配列Aに格納される。図14において、 $\alpha(n1, j2)$ の実部 $Re(\alpha(n1, j2))$ はAの左半分に格納され、虚部 $Im(\alpha(n1, j2))$ はAの右半分に格納されている。

【0059】そして、各プロセッサ内で、(9)式に準

じて $\alpha(n1, j2)$ に回転因子を乗算し、その結果を $\alpha'(n1, j2)$ として再び配列Aに格納する(ステップS5)。図15は、 $\alpha'(n1, j2)$ を格納した配列Aを示している。図15においては、図14の場合と同様に、 $\alpha'(n1, j2)$ の実部 $\text{Re}(\alpha'(n1, j2))$ はAの左半分に格納され、虚部 $\text{Im}(\alpha'(n1, j2))$ はAの右半分に格納されている。

【0060】次に、プロセッサPE1~PE4は、行列Aを図7の処理によりブロックレベルで転置して、配列BR、BIに格納する(ステップS6)。このとき、Aの左半分と右半分はそれぞれ図8に示すように分割され、対角方向に並ぶ4つのブロックが転置されて並列転送される。このような並列転送を繰り返すことにより、行列全体が転置される。図16は、転置後のデータを格納した配列BR、BIを示している。ここでは、BR、BIはそれぞれ 8×24 の配列として利用され、各プロセッサはBR、BIのそれぞれの 2×24 の部分に格納している。例えば、図15のプロセッサPE1の1行目の行ベクトル $\text{Re}(\alpha'(1, 1)), \dots, \text{Re}(\alpha'(1, 8)), \text{Im}(\alpha'(1, 1)), \dots, \text{Im}(\alpha'(1, 8))$ のうち、 $\text{Re}(\alpha'(1, 1)), \dots, \text{Re}(\alpha'(1, 8))$

$$f_{j1j2} = \sum_{n2=0}^{q-1} \sum_{n1=0}^{p-1} \omega_q^{n2j2} \omega_p^{n1j1} x_{n1n2} \quad \dots (27)$$

【0064】(27)式を(7)式と比べると、回転因子の乗算を除いて右辺の計算手順が同じであることが分かる。ただし、 x_{n1n2} は(7)式の $x(n1, n2)$ に対応し、 f_{j1j2} は $f(j2, j1)$ に対応している。したがって、並列計算機により(27)式を計算するには、図3から3段目の処理を除いた処理を行い、得られた2次元配列を転置すればよい。

【0065】図17は、 $p \times q$ 個の2次元データを(27)式に従って変換する2次元実FFTのフローチャートである。図17において処理が開始されると、まず出力装置13は与えられた $p \times q$ の2次元データを取り込み、これを $p \times q$ の2次元配列として、各プロセッサに所定数の行ベクトルを均等に割り当てる(ステップS21)。割当てられた行ベクトルは、図4と同様にし、各プロセッサのメモリ16に格納される。

【0066】次に、各プロセッサ内で、1次元実FFTの場合と同様にし、(8)式に対応する1段目の処理を並列に行う。まず、各プロセッサは行ベクトルのペアを作り、一方を実部、他方を虚部として、図5と同様の格納領域BR、BIにそれぞれコピーする(ステップS22)。次に、各プロセッサで、BRの行ベクトルと、対応するBIの行ベクトルとを束ねて、(13)式により複素数のフーリエ変換を行う(ステップS23)。そして、その結果から(16)~(19)式により、最初の $(q/2+1)$ 個の実フーリエ変換の結果を求め、その実部および虚部を、図6と同様にし、配列Aに格納す

はBRの1列目に格納され、 $\text{Im}(\alpha'(1, 1)), \dots, \text{Im}(\alpha'(1, 8))$ はBIの1列目に格納される。

【0061】次に、各プロセッサ内で転置後の行ベクトルを使用して、 $\alpha'(n1, j2)$ ($n1=1, \dots, 20$)の $n1$ に関するフーリエ変換を計算する(ステップS7)。このときの計算方法は、ステップS3の場合と同様である。計算に必要な $\alpha'(n1, j2)$ の実部と虚部は、それぞれBRとBI内の対応する位置に格納されている。そして、各行ベクトルについて得られた20個のフーリエ変換の結果は、実部と虚部に分けてそれぞれBRとBIに格納される。こうして、最終的に、図11の 20×15 個の実数に対する1次元実FFTの結果のうち、最初の約半分が求められる。残りの部分は、得られた結果の複素共役を取ることで求められることができる。

【0062】次に、2次元実FFTの方法について説明する。 $p \times q$ の2次元データの離散フーリエ変換は次式で与えられる。

【0063】

【数6】

る(ステップS24)。

【0067】次に、2段目の処理を行って、配列Aの $p \times (q/2+1) \times 2$ のデータをブロック(小領域)に分割し、ブロックレベルで転置して配列Bに格納する(ステップS25)。このときの転置処理の方法は、図7~10に示す方法と同様である。

【0068】そして、各プロセッサで3段目の処理を並列に行って、行列の転置結果における長さ p の各行ベクトルに対する複素FFTを計算する(ステップS26)。この結果得られた複素数の実部と虚部は、それぞれBR($1: q/2+1, p$)とBI($1: q/2+1, p$)に格納される。

【0069】次に、BRとBIに格納された配列をブロックレベルで再び転置して、配列Aに格納し(ステップS27)、処理を終了する。このとき、ステップS25と同様のデータ転送により、BR($1: q/2+1, p$)のデータがA($p, 1: q/2+1$)に転置され、BI($1: q/2+1, p$)のデータがA($p, q/2+2: (q/2+1) \times 2$)に転置される。

【0070】こうして、与えられた $p \times q$ 個の実数のフーリエ変換の結果のうち、 $p \times (q/2+1)$ 個が得られる。2次元実フーリエ変換の結果にも1次元実フーリエ変換の場合と同様の共役関係があるため、残りの $p \times (q - (q/2+1))$ 個の結果は得られた結果から求められる。

【0071】次に、2次元実FFTの拡張として、3次

元実FFTの方法について説明する。 $p \times q \times r$ の3次元データの離散フーリエ変換は次式で与えられる。

$$f_{j_1 j_2 j_3} = \sum_{n_3=0}^{r-1} \sum_{n_2=0}^{q-1} \sum_{n_1=0}^{p-1} \omega_r^{n_3 j_3} \omega_q^{n_2 j_2} \omega_p^{n_1 j_1} x_{n_1 n_2 n_3}$$

… (28)

【0073】(28)式は、(27)式の3次元への拡張となっているため、3次元実FFTは基本的に2次元実FFTと同様にして行う。ただし、増えた次元に相当するフーリエ変換処理が追加される。

【0074】図18は、 $p \times q \times r$ 個の3次元データを(28)式に従って変換する3次元実FFTのフローチャートである。図18において処理が開始されると、まず入出力装置13は与えられた3次元データを取り込み、それを $p \times q \times r$ の3次元配列とする。そして、この3次元配列を第1次元で分割して、各プロセッサに所

$$s_1 = 2 \times \text{proc} \times k_1,$$

$$k_1 = (p + 2 \times \text{proc} - 1) / (2 \times \text{proc}) \quad \dots (29)$$

と定め、第1次元を proc 台のプロセッサにより均等に分割したとき、各プロセッサに格納される部分の行数

$$k_4 = (r / 2 + 1) \times 2$$

以上とする。図19では、Aの第3次元の大きさは k_4 となっており、配列Aは $A(s_1, q, k_4)$ と表され

$$k_5 = ((r / 2 + \text{proc}) / \text{proc}) \times \text{proc} \quad \dots (30)$$

として、Aの第1次元と第3次元の転置用の格納配列B($2 \times k_5, q, s_1$)を用意する。Bは同じ大きさの2つの配列BR(k_5, q, s_1)とBI(k_5, q, s_1)からなる。BR、BIの第1次元の大きさ k_5

$$2 \times k_5 \geq k_4 = (q / 2 + 1) \times 2$$

なる関係がある。

【0076】そして、A(s_1, q, k_4)、BR(k_5, q, s_1)、BI(k_5, q, s_1)の第1次元を各プロセッサで均等に分割し、分割された各プロセッサの部分 $a(2 \times k_1, q, k_4)$ 、 $b_r(k_5 / \text{proc}, q, s_1)$ 、 $b_i(k_5 / \text{proc}, q, s_1)$ とする。

【0077】次に、各プロセッサ内で、1段目の処理を並列に行う。ただし、このときメモリの使用領域を節約するために、各プロセッサに割り当てられた第3次元のベクトルを前半と後半の2組に分け、一方を実部、他方を虚部として、複素数のフーリエ変換を行う。

【0078】まず、各プロセッサは第3次元のベクトルのペアを作り、一方を実部、他方を虚部として別々の領域に格納する(ステップS32)。具体的には、格納領域 b_r 、 b_i をそれぞれ格納配列 $b_r(k_1, q, 2 \times k_5)$ 、 $b_i(k_1, q, 2 \times k_5)$ として利用し、 $a(1 : k_1, q, k_4)$ を $b_r(k_1, q, 1 : k_4)$ にコピーし、 $a(k_1 + 1 : 2 \times k_1, q, k_4)$ を $b_i(k_1, q, 1 : k_4)$ にコピーする。

【0072】

【数7】

定数の第3次元のベクトルを均等に割り当てる(ステップS31)。割当てられた第3次元のベクトルは、各プロセッサのメモリ16に格納される。

【0075】図19は、このときの3次元データを格納する3次元格納配列Aの分割方法を示している。図19では、簡単のためプロセッサ数を4(PE1、PE2、PE3、PE4)としているが、より一般的にはそれを proc と書くことにする。このとき、Aの第1次元(行の次元)の大きさ s_1 を、(22)、(23)式と同様に、

$$\dots (29)$$

$$\dots (30)$$

が偶数になるようにする。また、Aの第2次元の大きさは q とし、第3次元の大きさは、

$$\dots (31)$$

る。次に、

は、Aの第3次元の大きさ k_4 の半分を proc 台のプロセッサで分割できるように、 $k_4 / 2 = (r / 2 + 1)$ を修正したものである。したがって、

$$\dots (33)$$

【0079】図19では、4台のプロセッサPE1、PE2、PE3、PE4の各々に割り当てられた $a(2 \times k_1, q, k_4)$ が、前半部分と後半部分に分割されている。PE1のデータはR1とI1に分けられ、PE2のデータはR2とI2に分けられ、PE3のデータはR3とI3に分けられ、PE4のデータはR4とI4に分けられる。そして、これらのデータが、図20に示すように、配列BR、BIにコピーされる。図20において、R1、R2、R3、R4は実部としてBRに格納され、I1、I2、I3、I4は虚部としてBIに格納されている。このとき、BR、BIは、それぞれBR($s_1 / 2, q, 2 \times k_5$)、BI($s_1 / 2, q, 2 \times k_5$)なる配列として利用される。

【0080】次に、 $b_r(k_1, q, 1 : k_4)$ の第3次元のベクトルと、対応する $b_i(k_1, q, 1 : k_4)$ の第3次元のベクトルとを束ねて、各プロセッサで複素数のフーリエ変換を行う(ステップS33)。つまり、 $b_r(k_1, q, 1 : k_4)$ の1つの第3次元方向のベクトルの各要素を(12)式の x_{1n} とみなし、対応する $b_i(k_1, q, 1 : k_4)$ の第3次元方向のベ

クトルの要素を(12)式の x_{2n} とみなして、複素数 z_n に対するフーリエ変換を(13)式により計算する。ただし、このとき $Q=r$ とする。

【0081】そして、(16)～(19)式により、最初の $(r/2+1)$ 個の実フーリエ変換の結果を求め、その実部および虚部をそれぞれ $a(2 \times k1, q, 1:r/2+1)$ および $a(2 \times k1, q, r/2+2:k4)$ に格納する(ステップS34)。図21は、このときの格納方法を示している。図21において、Aの第3次元の前半が実部に割り当てられ、後半が虚部に割り当てられていることが分かる。例えばプロセッサPE1内では、図20のR1部分のベクトルの実フーリエ変換結果のうち、最初の $(r/2+1)$ 個が実際に求められ、それらの実部がCR1に格納され、虚部がDR1に格納される。また、図20のI1部分のベクトルの実フーリエ変換結果のうち、最初の $(r/2+1)$ 個が実際に求められ、それらの実部がCI1に格納され、虚部がDI1に格納される。他のプロセッサの領域CR2、DR2、CI2、DI2、CR3、DR3、CI3、DI3、CR4、DR4、CI4、DI4についても同様である。

【0082】次に、各プロセッサ内で、 $a(2 \times k1, q, 1:r/2+1)$ の第2次元方向のベクトルと、 $a(2 \times k1, q, r/2+2:k4)$ の第2次元方向のベクトルとを束ねて、長さ q の複素数のフーリエ変換を行う(ステップS35)。そして、その結果の実部および虚部を、それぞれ $a(2 \times k1, q, 1:r/2+1)$ および $a(2 \times k1, q, r/2+2:k4)$ に格納する。

【0083】次に、2段目の処理を行って、配列Aの $p \times q \times (r/2+1) \times 2$ のデータをブロックに分割し、ブロックレベルで第1次元と第3次元の間の転置を行い、配列Bに格納する(ステップS36)。このとき、格納領域BR、BIはそれぞれ配列BR($k5, q, s1$)、BI($k5, q, s1$)として利用され、その第1次元に関して分割されている。そして、プロセッサ間のデータ転送により、A($p, q, 1:r/2+1$)のデータがBR($1:r/2+1, q, p$)に転置され、A($p, q, r/2+2:(r/2+1) \times 2$)のデータがBI($1:r/2+1, q, p$)に転置される。この時行われる転置処理は、図7と同様である。ただし、ここでは図7における B_{KJ} 、 A_{JK}^T は、3次元配列を第1次元および第3次元に関して分割したブロックを意味する。図22は、転置前の配列Aの実部または虚部の分割例を示しており、図23は、転置後のデータを格納する配列BRまたはBIの分割例を示している。

【0084】図22において、Aの実部(または虚部)のうちプロセッサPE1内にあるデータは、A11、A12、A13、A14の4つのブロックに分割されている。同様に、プロセッサPE2内のデータはブロックA21、

A22、A23、A24に分割され、プロセッサPE3内のデータはブロックA31、A32、A33、A34に分割され、プロセッサPE4内のデータはブロックA41、A42、A43、A44に分割される。また、図23において、BR(またはBI)のうちプロセッサPE1内にあるデータは、B11、B12、B13、B14の4つのブロックに分割されている。同様に、プロセッサPE2内のデータはブロックB21、B22、B23、B24に分割され、プロセッサPE3内のデータはブロックB31、B32、B33、B34に分割され、プロセッサPE4内のデータはブロックB41、B42、B43、B44に分割される。

【0085】図24は、転置処理を行った後のBR(またはBI)の内容を示している。図24において、プロセッサPE1内には A_{11}^T 、 A_{21}^T 、 A_{31}^T 、 A_{41}^T が格納され、プロセッサPE2内には A_{12}^T 、 A_{22}^T 、 A_{32}^T 、 A_{42}^T が格納され、プロセッサPE3内には A_{13}^T 、 A_{23}^T 、 A_{33}^T 、 A_{43}^T が格納され、プロセッサPE4内には A_{14}^T 、 A_{24}^T 、 A_{34}^T 、 A_{44}^T が格納されている。

【0086】3次元の場合も、図8に示す2次元配列のデータ転送と同様に、図22の配列の対角方向に並ぶブロック要素について並列にデータ転送を行う。例えば、図22の斜線部分のブロックA12、A23、A34、A41の転置データ A_{12}^T 、 A_{23}^T 、 A_{34}^T 、 A_{41}^T は、図24の斜線部分の各ブロックに並列に転送される。

【0087】そして、各プロセッサで3段目の処理を並列に行って、配列の転置結果における長さ p の各第3次元のベクトルに対する複素FFTを計算する(ステップS37)。この結果得られた複素数の実部と虚部は、それぞれBR($1:r/2+1, q, p$)とBI($1:r/2+1, q, p$)に格納される。

【0088】次に、BRとBIに格納された配列の第1次元と第3次元の間の転置を、再びブロックレベルで行って配列Aに格納し(ステップS38)、処理を終了する。このとき、ステップS36と同様のデータ転送により、BR($1:r/2+1, q, p$)のデータがA($p, q, 1:r/2+1$)に転置され、BI($1:r/2+1, q, p$)のデータがA($p, q, r/2+2:(r/2+1) \times 2$)に転置される。

【0089】こうして、与えられた $p \times q \times r$ 個の実数のフーリエ変換の結果のうち、 $p \times q \times (r/2+1)$ 個が得られる。3次元実フーリエ変換の結果にも1次元実フーリエ変換の場合と同様の共役関係があるため、残りの $p \times q \times (r - (r/2+1))$ 個の結果は得られた結果から求められる。

【0090】以上説明したように、本実施例では、ベクトル計算機向けの1～3次元実フーリエ変換を、2つの実数列をそれぞれ実部および虚部とみなした複素フーリエ変換に置き換え、並列計算機の各プロセッサでその結果の約半分を取り出して、最終的にすべての結果を求め

ている。尚、本発明は、1～3次元実フーリエ変換に限らず、より高い次元の離散フーリエ変換にも容易に拡張することができる。例えば、D次元の実フーリエ変換を計算するには、図18の3次元実フーリエ変換の処理のステップS35において、(D-2)次元部分の複素フーリエ変換を続けて行えばよい。

【0091】

【発明の効果】本発明によれば、メモリ分散型並列計算機システムによる実フーリエ変換処理において、変換される実数配列の1つの次元の大きさが各プロセッサ内で偶数になるように、データが分割配置される。これにより、各プロセッサ内で実数ベクトルのペアを作ることが可能になり、一方を実部、他方を虚部とみなして、他の次元についてのフーリエ変換をそのプロセッサ内で閉じて行うことができる。したがって、並列計算機の並列性およびベクトル性能を効率良く利用できる。

【0092】また、並列転置と組み合わせることでデータ転送のコストがさらに削減され、処理が効率化される。

【図面の簡単な説明】

【図1】本発明の原理図である。

【図2】実施例の並列計算機システムの構成図である。

【図3】1次元実FFTのフローチャートである。

【図4】2次元配列の分割を示す図である。

【図5】2次元配列のコピー領域を示す図である。

【図6】実フーリエ変換の結果を2次元配列に格納する方法を示す図である。

【図7】行列の転置処理のフローチャートである。

【図8】転置前の2次元配列を示す図である。

【図9】転置後の2次元配列を示す図である。

【図10】転置後の2次元配列の内容を示す図である。

【図11】2次元配列の例を示す図である。

【図12】コピーされた2次元配列の例を示す図である。

る。

【図13】複素フーリエ変換後の2次元配列の例を示す図である。

【図14】実フーリエ変換の結果の格納例を示す図である。

【図15】ローテーション計算後の2次元配列の例を示す図である。

【図16】転置後の2次元配列の例を示す図である。

【図17】2次元実FFTのフローチャートである。

【図18】3次元実FFTのフローチャートである。

【図19】3次元配列の分割を示す図である。

【図20】3次元配列のコピー領域を示す図である。

【図21】実フーリエ変換の結果を3次元配列に格納する方法を示す図である。

【図22】転置前の3次元配列を示す図である。

【図23】転置後の3次元配列を示す図である。

【図24】転置後の3次元配列の内容を示す図である。

【符号の説明】

1 入力手段

2 配列記憶手段

3 第1の記憶手段

4 第2の記憶手段

5 計算手段

6 転置手段

7 出力手段

11-1, 11-2, 11-M プロセッサ

12 ネットワーク

13 入出力装置

14 通信部

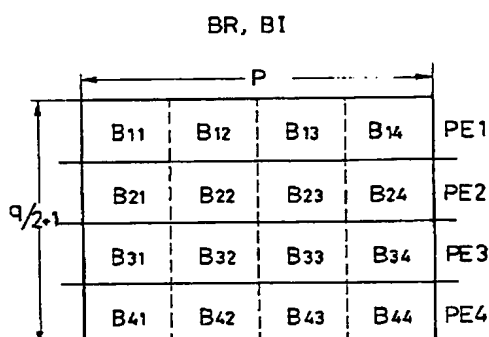
15 処理部

16 メモリ

17 内部バス

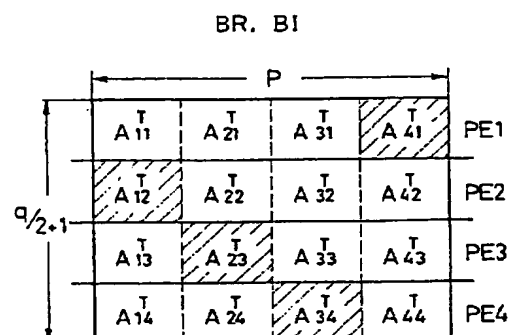
【図9】

転置後の2次元配列を示す図



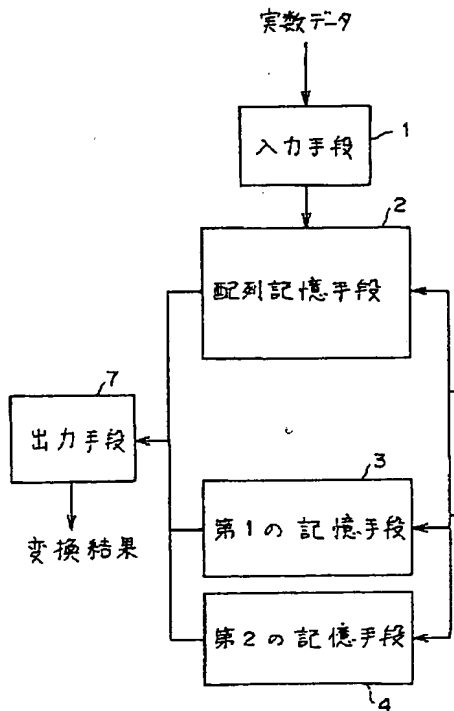
【図10】

転置後の2次元配列の内容を示す図



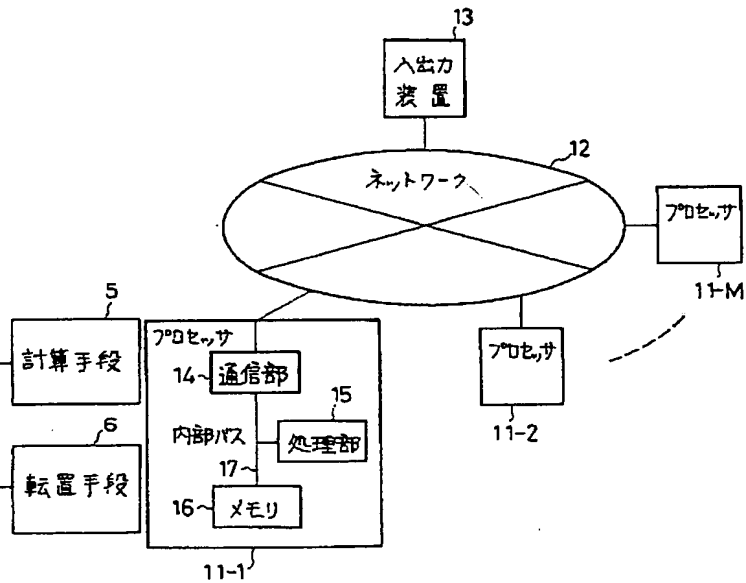
【図 1】

本発明の原理図



【図 2】

実施例のシステム構成図

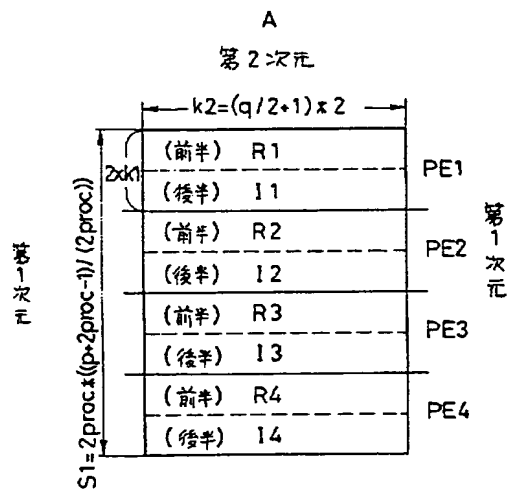


【図 8】

転置前の 2 次元配列を示す図

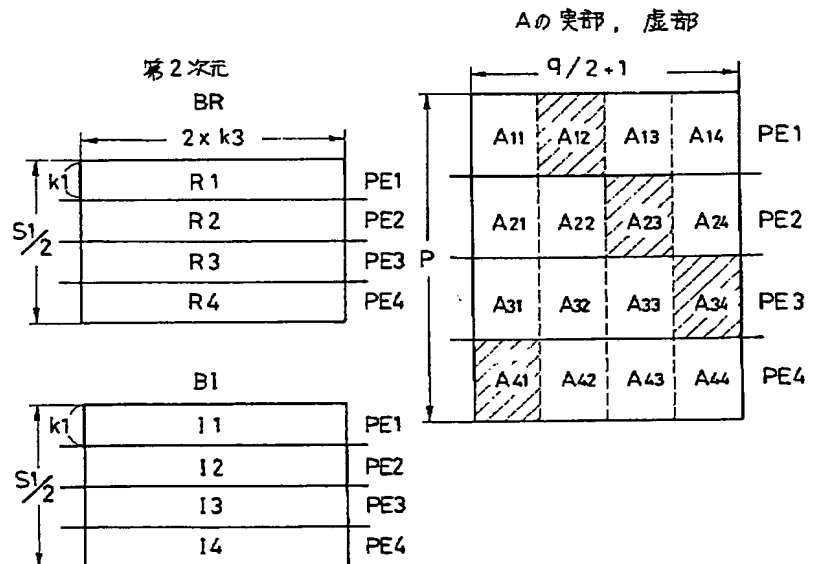
【図 4】

2次元配列の分割を示す図



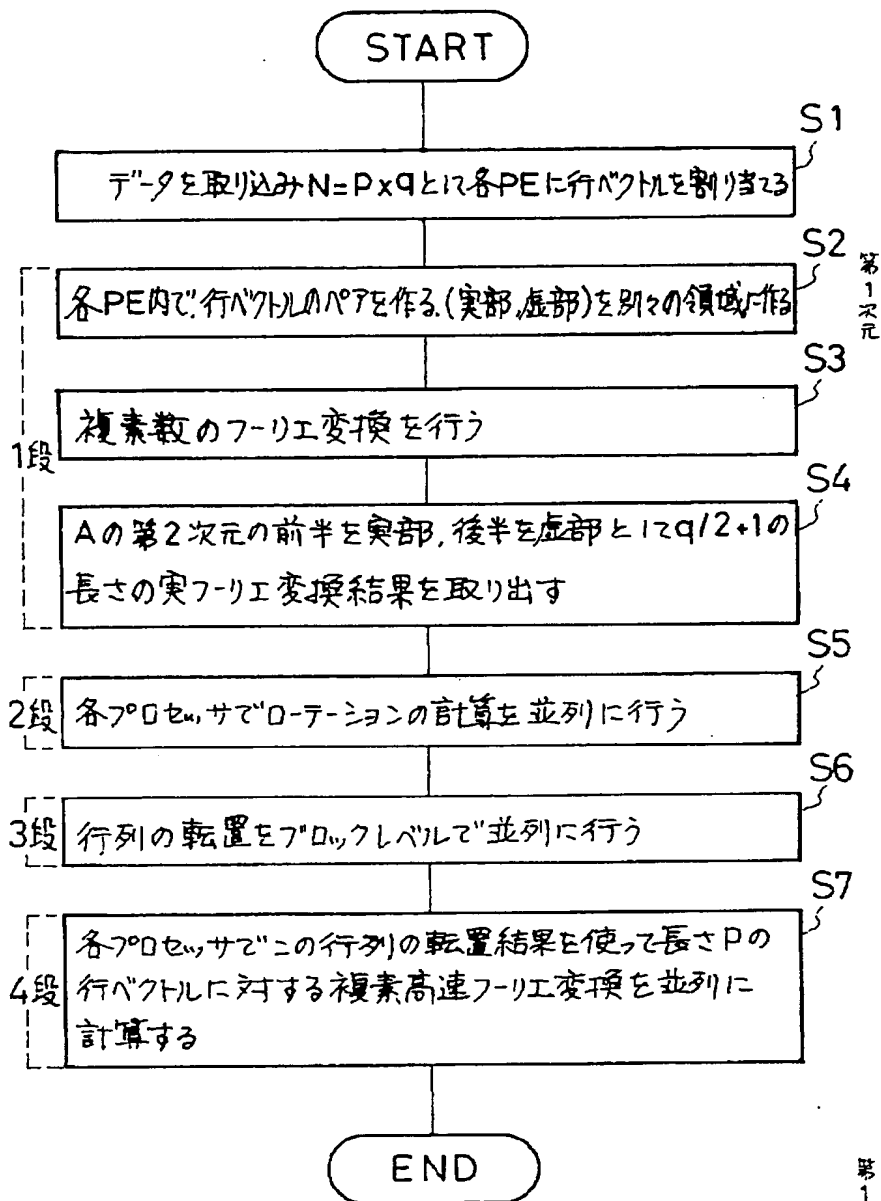
【図 5】

2次元配列のコピー領域を示す図



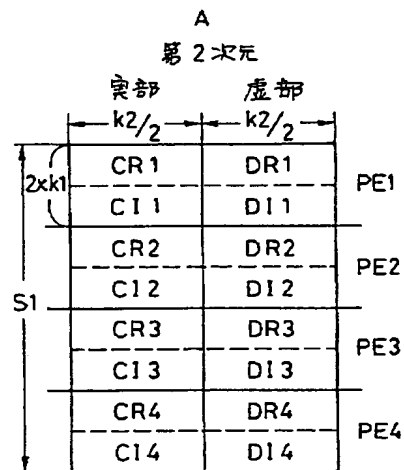
【図 3】

一次元実FFTのフローチャート



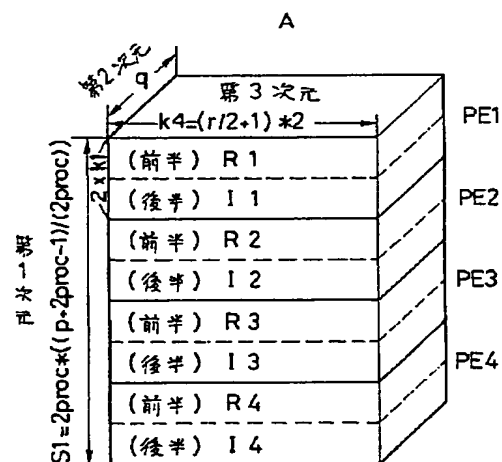
【図 6】

実フーリエ変換の結果を2次元配列に格納する方法を示す図



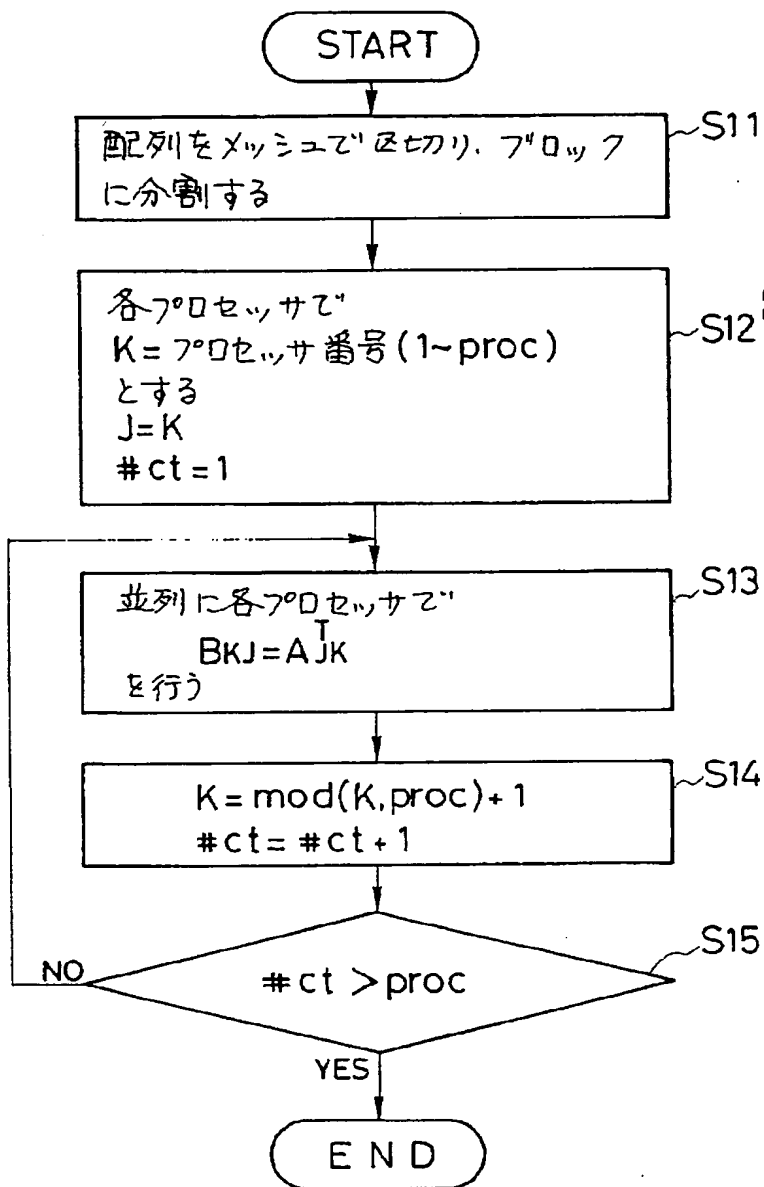
【図 19】

3次元配列の分割を示す図



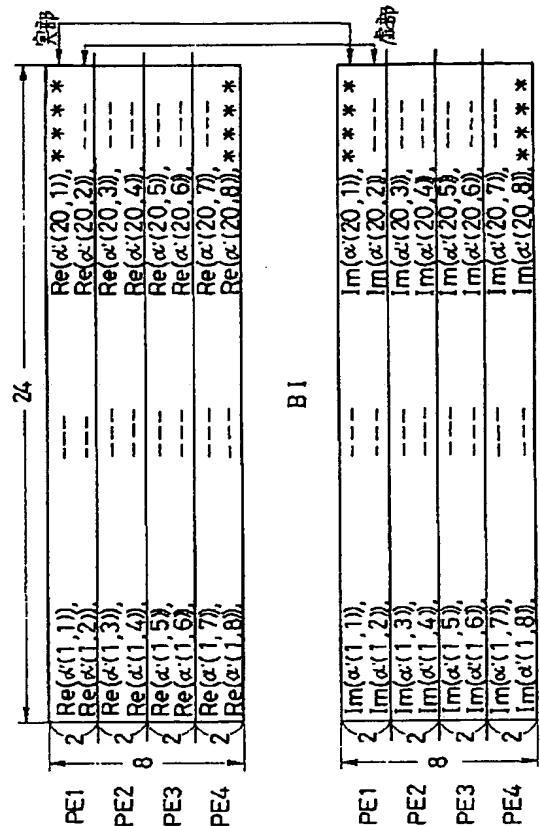
【図7】

行列の転置処理のフローチャート



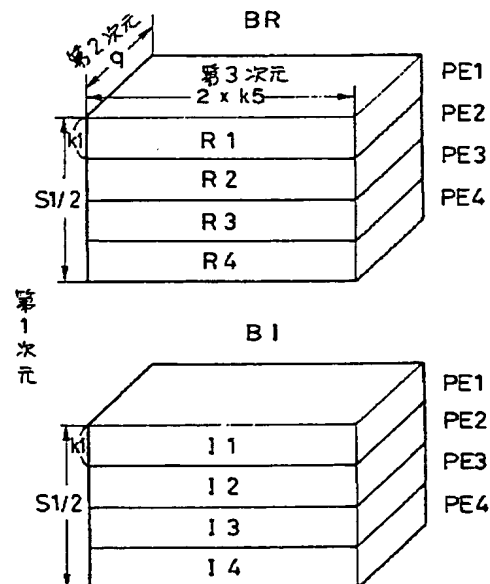
【図16】

転置後の2次元配列の例を示す図



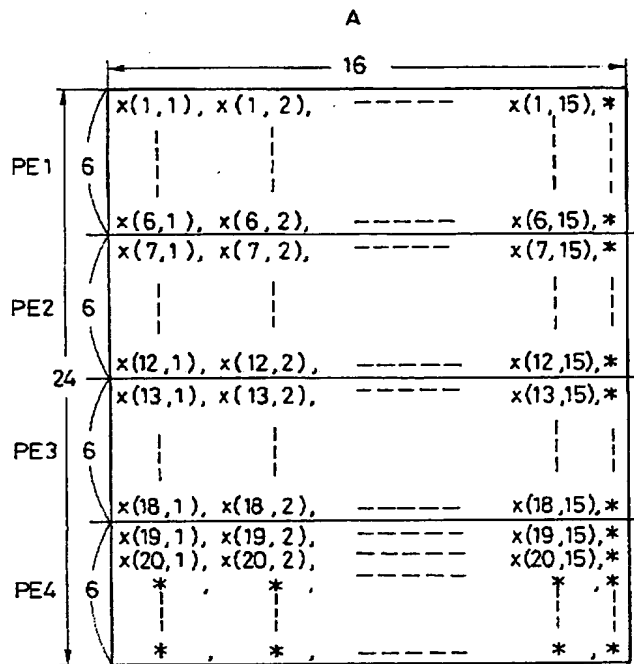
【図20】

3次元配列のコピー領域を示す図



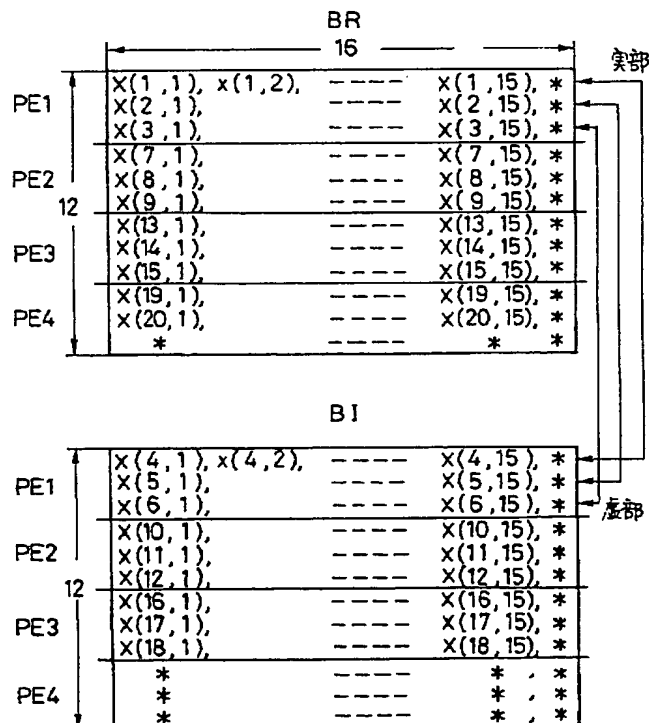
【図 1 1】

2次元配列の例を示す図



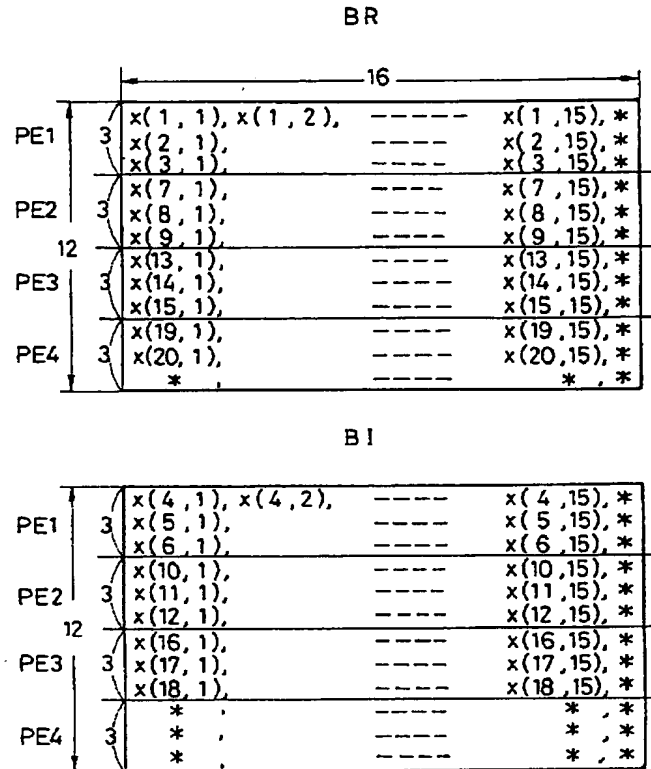
【図 1 3】

複素フーリエ変換後の2次元配列の例を示す図



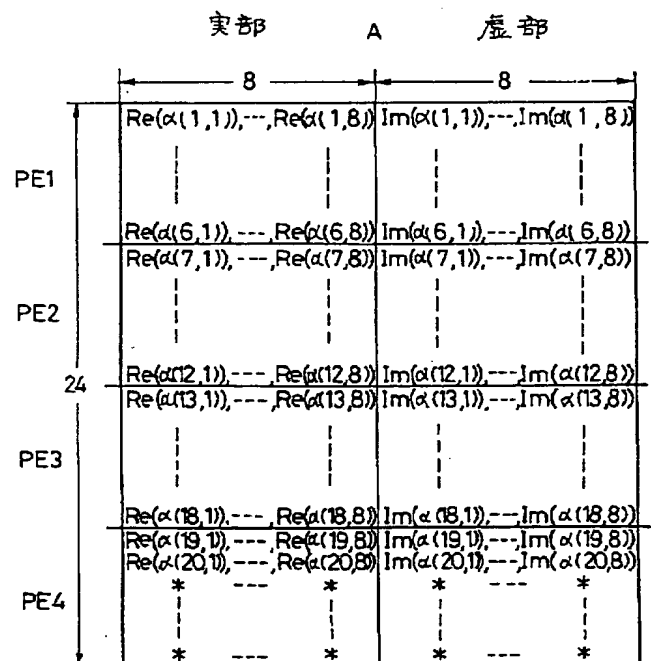
【図 1 2】

コピーされた2次元配列の例を示す図



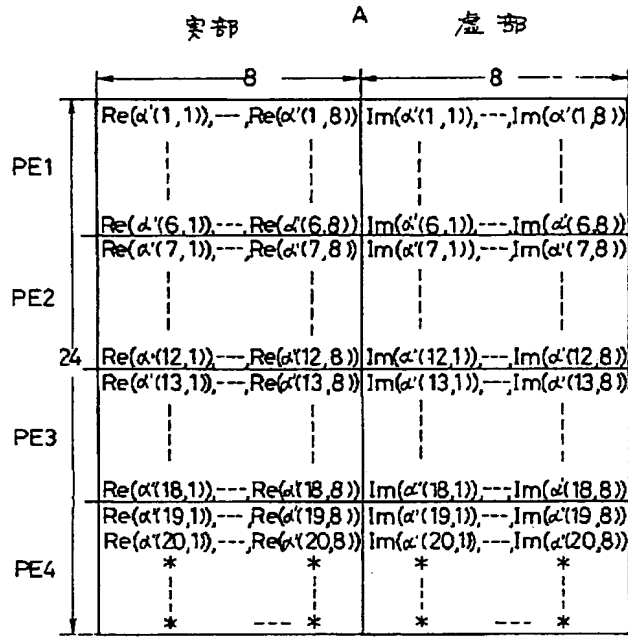
【図 1 4】

実フーリエ変換の結果の格納例を示す図



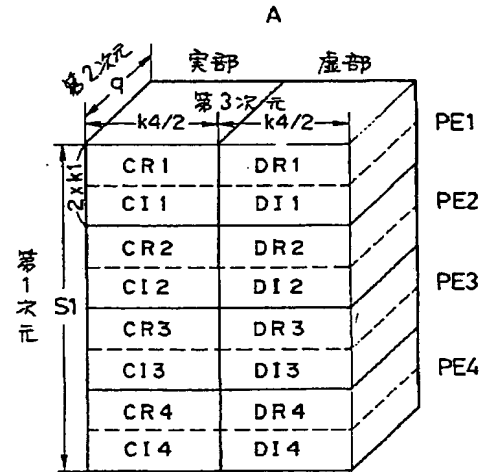
【図 15】

ローテーション計算後の2次元配列の例を示す図



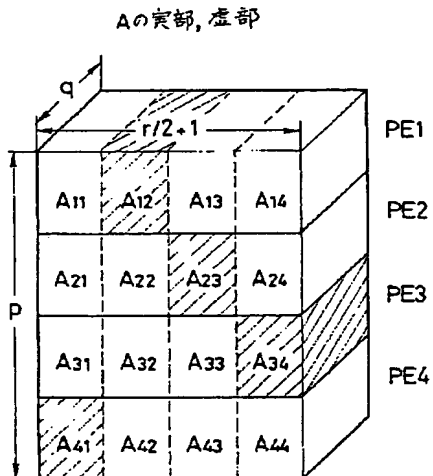
【図 21】

実フーリエ変換の結果を3次元配列に格納する方法を示す図



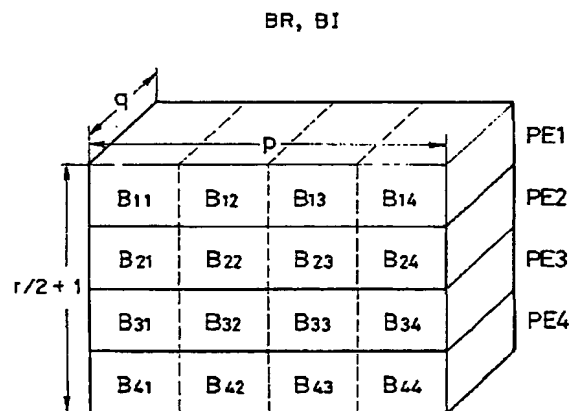
【図 22】

転置前の3次元配列を示す図



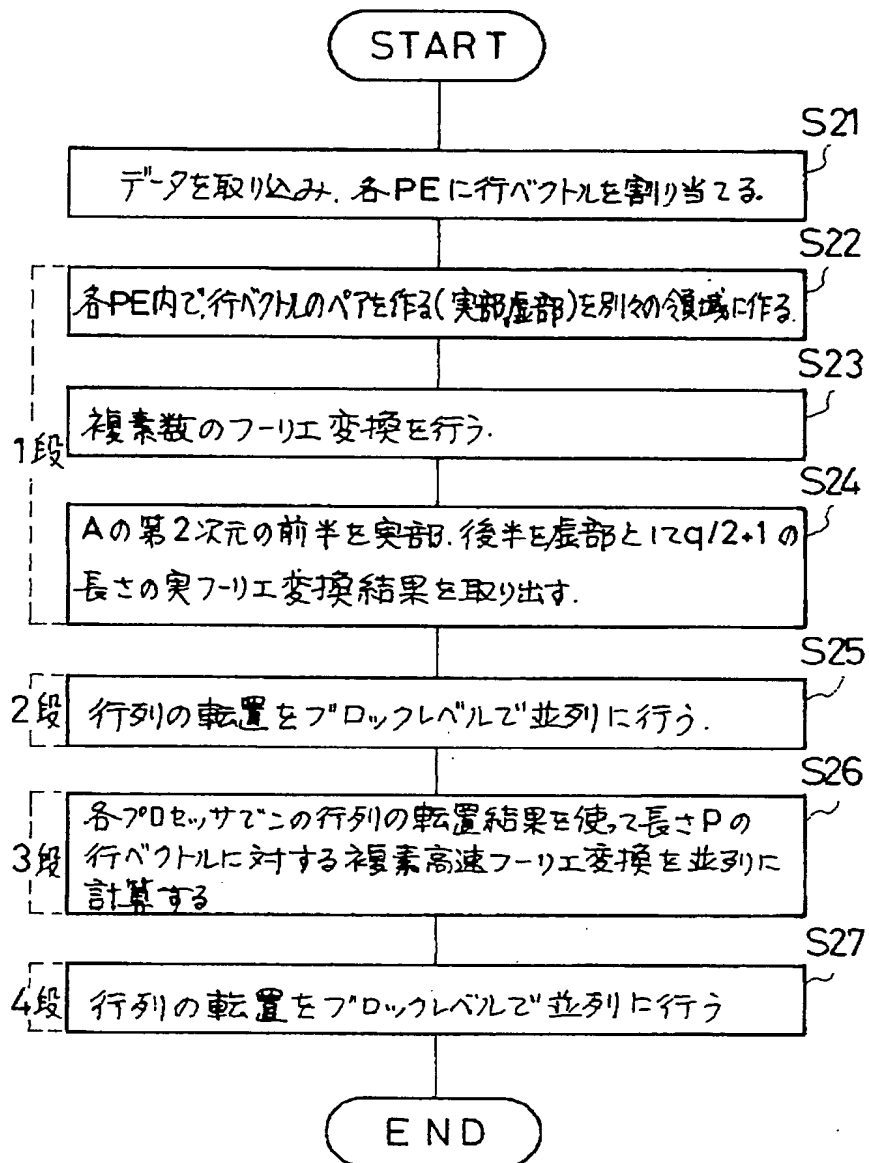
【図 23】

転置後の3次元配列を示す図



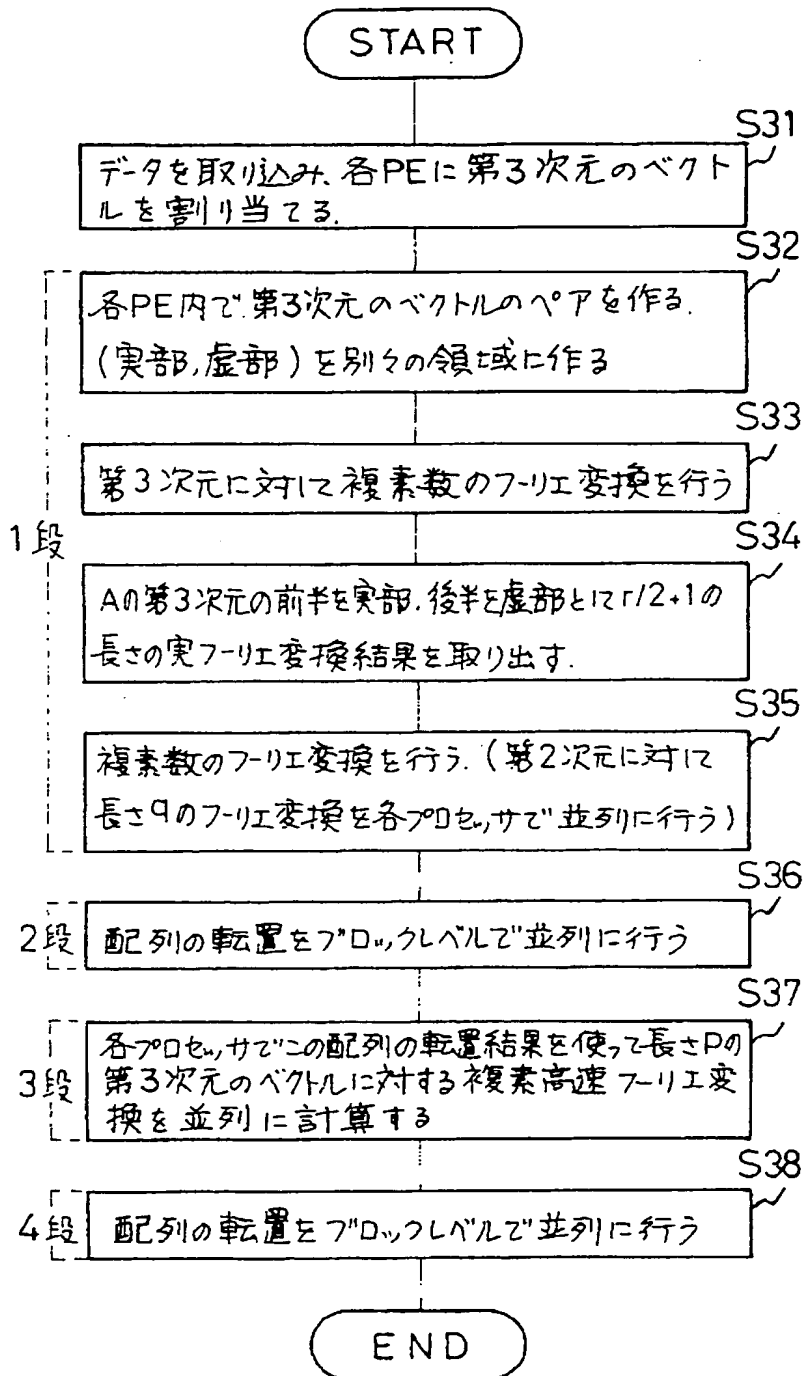
【図17】

2次元実FFTのフローチャート



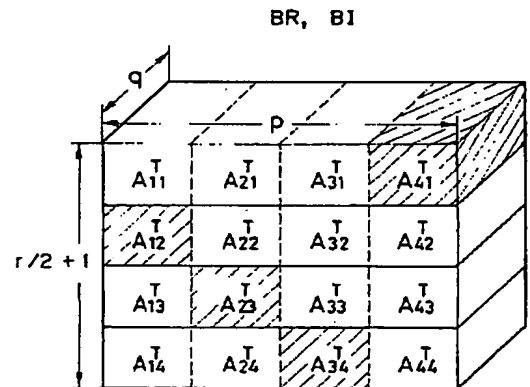
【図18】

3次元実FFTのフローチャート



【図24】

転置後の3次元配列の内容を示す図



フロントページの続き

(72)発明者 マーカス ヘグランド
オーストラリア国、エーシーティ、2601、
オコーナー、バートストリート 9 番地

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKewed/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.